OPT2025: 17th Annual Workshop on Optimization for Machine Learning

Understanding and Improving Shampoo via Kullback–Leibler Minimization

Wu Lin YORKER.LIN@GMAIL.COM
Scott C. Lowe SCOTT.LOWE@VECTORINSTITUTE.AI
Felix Dangel FDANGEL@VECTORINSTITUTE.AI
Vector Institute, Canada

Runa Eschenhagen RE393@CAM.AC.UK
University of Cambridge, United Kingdom

Zikun Xu ZIKUNXU@MICROSOFT.COM
Microsoft, United States

Roger B. Grosse RGROSSE@CS.TORONTO.EDU
University of Toronto & Vector Institute, Canada

Abstract

As an adaptive method, Shampoo employs a structured second-moment estimation, and its effectiveness has attracted growing attention. Prior work has primarily analyzed its estimation scheme through the Frobenius norm. Motivated by the natural connection between the second moment and a covariance matrix, we propose studying Shampoo's estimation as covariance estimation through the lens of Kullback-Leibler (KL) minimization. This alternative perspective reveals a previously hidden limitation, motivating improvements to Shampoo's design. Building on this insight, we develop a practical estimation scheme, termed KL-Shampoo, that eliminates Shampoo's reliance on Adam for stabilization, thereby removing the additional memory overhead introduced by Adam. Preliminary results show that KL-Shampoo improves Shampoo's performance, enabling it to stabilize without Adam and even outperform its Adam-stabilized variant, SOAP, in neural network pretraining.

1. Introduction

Shampoo [13] has received significant attention [4, 5, 11, 21, 24, 28] due to its strong performance in training a wide range of neural network (NN) models [8, 15]. A deeper understanding of this method could help unlock its full potential. Prior work [4, 11, 21, 28] has investigated the structural preconditioner scheme of Shampoo—which approximate the full-matrix gradient 2^{nd} moment [9]—through the Frobenius norm. Few studies, however, have examined Shampoo's scheme from the perspective of Kullback–Leibler (KL) divergence. Compared to the Frobenius norm, the KL divergence is more suitable [3, 20] for viewing its scheme as a covariance estimation scheme, since the gradient 2^{nd}-moment it approximates can be interpreted as a covariance matrix. Moreover, this divergence naturally respects the symmetric positive-definite (SPD) constraint [6, 23] implicitly imposed on Shampoo's preconditioner whereas the Frobenius norm does not. This constraint is crucial: Shampoo requires its preconditioner to be SPD to ensure that the preconditioned gradient direction is a descent direction [22].

In this work, we introduce a KL perspective that interprets Shampoo's estimation scheme as the solution to a KL minimization problem. This perspective reveals a key limitation of Shampoo's estimation that remains hidden under the Frobenius-norm interpretation and opens new opportunities for improvement. Unlike existing interpretations, which focus primarily on matrix-valued weights, our approach extends naturally to tensor-valued settings. Leveraging this perspective, we refine the design of Shampoo's estimation and develop a practical KL-based scheme, termed **KL-Shampoo**, for training neural networks (NNs). Importantly, KL-Shampoo eliminates the need for step-size grafting with Adam [2], as required for stabilizing Shampoo [5, 11, 24]. Preliminary results show that KL-Shampoo is both effective and stable for training NNs, including NNs with tensor-valued weights, outperforming both Shampoo with step-size grafting and an Adam-stabilized variant—SOAP [25].

2. Background

Notation For notational simplicity, we focus on matrix-valued weights and consider a single weight matrix $\boldsymbol{\Theta} \in \mathcal{R}^{d_a \times d_b}$, rather than a set of weight matrices for training. We use $\mathrm{Mat}(\cdot)$ to unflatten its input vector into a matrix and $\mathrm{vec}(\cdot)$ to flatten its input matrix into a vector. For example, $\boldsymbol{\theta} := \mathrm{vec}(\boldsymbol{\Theta})$ is the flattened weight vector and $\boldsymbol{\Theta} \equiv \mathrm{Mat}(\boldsymbol{\theta})$ is the original (unflattened) weight matrix. Vector $\boldsymbol{g}$ is a (flattened) gradient vector for the weight matrix. We denote γ, β_2 and $\boldsymbol{S}$ to be a step size, a weight for moving average, and a preconditioning matrix for an adaptive method, respectively. $\mathrm{Diag}(\cdot)$ returns a diagonal matrix whose diagonal entries are given by its input vector, whilst $\mathrm{diag}(\cdot)$ extracts the diagonal entries of its input matrix as a vector.

Shampoo Given a matrix gradient $\boldsymbol{G} := \mathrm{Mat}(\boldsymbol{g})$, the original Shampoo method [13] considers a *Kronecker-factored* approximation, $(\boldsymbol{S}_a)^{2p} \otimes (\boldsymbol{S}_b)^{2p}$, of the full-matrix gradient second moment, $\mathbb{E}_{\boldsymbol{g}}[\boldsymbol{g}\boldsymbol{g}^\top]$, where p denotes a matrix power, $\boldsymbol{S}_a := \mathbb{E}_{\boldsymbol{g}}[\boldsymbol{G}\boldsymbol{G}^\top]$, $\boldsymbol{S}_b := \mathbb{E}_{\boldsymbol{g}}[\boldsymbol{G}^\top\boldsymbol{G}]$, and $\otimes$ denotes a Kronecker product. In mini-batch settings, we often approximate the expectation with just one gradient outer product such as $\boldsymbol{g}\boldsymbol{g}^\top \approx \mathbb{E}_{\boldsymbol{g}}[\boldsymbol{g}\boldsymbol{g}^\top]$ [21]. The original shampoo method uses the $1/4$ power (i.e., $p = 1/4$) and other works [5, 21, 24] suggest using the $1/2$ power (i.e., $p = 1/2$). At each iteration, Shampoo follows this update rule:

$$\begin{aligned} &\boldsymbol{S}_a \leftarrow (1-\beta_2)\boldsymbol{S}_a + \beta_2 \boldsymbol{G}\boldsymbol{G}^\top, \quad \boldsymbol{S}_b \leftarrow (1-\beta_2)\boldsymbol{S}_b + \beta_2 \boldsymbol{G}^\top\boldsymbol{G} \quad \text{(Kronecker 2}^{\text{nd}}\text{ moment est.)}, \\ &\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \boldsymbol{S}^{-1/2}\boldsymbol{g} \iff \boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \gamma \boldsymbol{S}_a^{-p}\boldsymbol{G}\boldsymbol{S}_b^{-p} \quad \text{(preconditioning)}, \end{aligned} \tag{1}$$

where $\boldsymbol{S} := \boldsymbol{S}_a^{2p} \otimes \boldsymbol{S}_b^{2p}$ is Shampoo's preconditioning matrix, and we leverage the Kronecker structure of $\boldsymbol{S}$ to move from the left expression to the right expression in the second line.

Shampoo's estimation rule is not motivated as covariance estimation. The original Shampoo's Kronecker estimation rule ($p = 1/4$) [10, 13] is proposed based on a matrix Loewner bound [19], while recent estimation rules ($p = 1/2$) [11, 21] focus on bounds induced by the Frobenius norm. Neither of these sets of works interpret the estimation as covariance estimation.

Shampoo's implementation employs eigendecomposition. Because computing a matrix p-power at each step is expensive, Shampoo is implemented [5, 24] by using the eigendecomposition of $\boldsymbol{S}_k$, such as $\boldsymbol{Q}_k \mathrm{Diag}(\boldsymbol{\lambda}_k)\boldsymbol{Q}_k^\top = \mathrm{eigen}(\boldsymbol{S}_k)$, for $k \in \{a, b\}$, every few steps and storing eigenfactors $\boldsymbol{Q}_k$ and $\boldsymbol{\lambda}_k$. Therefore, the power of $\boldsymbol{S}_k$ is computed using an elementwise power in $\boldsymbol{\lambda}_k$ such as $\boldsymbol{S}_k^{-p} = \boldsymbol{Q}_k \mathrm{Diag}\big(\boldsymbol{\lambda}_k^{\odot -p}\big)\boldsymbol{Q}_k^\top$, where $^{\odot p}$ denotes elementwise pth power. This computation becomes an approximation if the decomposition is not performed at every step.

Using Adam for Shampoo's stabilization increases memory usage. If the eigendecomposition is applied infrequently to reduce iteration cost, Shampoo has to apply step-size grafting with Adam to maintain performance [1, 24]. Unfortunately, this increases its memory usage.

3. Second Moment Estimation via Kullback–Leibler Minimization

We present a new perspective on interpreting and improving the second-moment estimation scheme of Shampoo, showing that this scheme can be viewed as a *structural covariance estimation* procedure via Kullback–Leibler (KL) minimization. This perspective reveals a key limitation of Shampoo's estimation rule that remains obscured under the conventional Frobenius-norm interpretation [4, 11, 21, 28], while also guiding the development of new, practical variants. Building on this insight, we propose a KL-based estimation scheme for Shampoo using QR decomposition, termed **KL-Shampoo**.

KL Minimization We begin by introducing a KL perspective for a matrix-valued case. For simplicity, we drop subscripts when referring to the flattened gradient 2nd moment, like $\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top] := \mathbb{E}_{\boldsymbol{g}}[\boldsymbol{g}\boldsymbol{g}^\top]$, where $\boldsymbol{g} = \mathrm{vec}(\boldsymbol{G})$ is a flattened gradient vector of a matrix-valued gradient $\boldsymbol{G} \in \mathcal{R}^{d_a \times d_b}$. The goal is to estimate a Kronecker-structured preconditioning matrix, $\boldsymbol{S} = \boldsymbol{S}_a \otimes \boldsymbol{S}_b$, that closely approximates the 2nd moment, where $\boldsymbol{S}_a \in \mathcal{R}^{d_a \times d_a}$ and $\boldsymbol{S}_b \in \mathcal{R}^{d_b \times d_b}$ are both symmetric positive-definite (SPD). Motivated by the natural connection between the second moment and a covariance matrix, we treat these as covariance matrices of zero-mean Gaussian distributions and achieve this goal by minimizing the KL divergence between the two distributions:

KL Perspective for Covariance Estimation

$$\begin{aligned}\mathrm{KL}(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}) &:= D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{0}, \mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top] + \kappa\boldsymbol{I}) \parallel \mathcal{N}(\boldsymbol{0}, \boldsymbol{S})) \\ &= \frac{1}{2}\big(\log\det(\boldsymbol{S}) + \mathrm{Tr}((\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top] + \kappa\boldsymbol{I})\boldsymbol{S}^{-1})\big) + \text{const}, \qquad (2)\end{aligned}$$

where $\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top]$ and $\boldsymbol{S}$ are considered as Gaussian's covariance, $\det(\cdot)$ denotes the determinant of its input, and $\kappa \geq 0$ is a damping weight to ensure the positive-definiteness of $\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top] + \kappa\boldsymbol{I}$ if necessary.

Justification of using the KL divergence Many existing works [4, 11, 21, 28] focus on matrix-valued weights and interpret Shampoo's estimation rule for such weights from the Frobenius-norm perspective. However, this norm does not account for the SPD constraint implicitly imposed on Shampoo's preconditioner $\boldsymbol{S}$ to ensure that the preconditioned direction is a descent direction [22]. As emphasized in the literature [6, 23], it is more appropriate to consider a "distance" that respects this constraint. We adopt the KL divergence because it naturally incorporates the SPD constraint, is widely used for covariance estimation [3, 20], and provides a unified framework to reinterpret and improve Shampoo's estimation rule, even for tensor-valued weights.

3.1. Interpreting Shampoo's estimation as covariance estimation

Similar to existing works [11, 21, 25], we disable the moving average (i.e., let $\beta_2 = 1$) for our descriptions and focus on Shampoo with power $p = 1/2$, presenting a KL minimization perspective and interpreting its estimation rule from this perspective. We will show that Shampoo's estimation rule can be obtained by solving a KL minimization problem.

Shampoo's estimation rule as Kronecker-based covariance estimation According to Claim 1, Shampoo's estimation rule with power $p = 1/2$ in Eq. (1) can be viewed as the optimal solution of a KL minimization problem (up to a constant scalar) when one Kronecker factor is updated

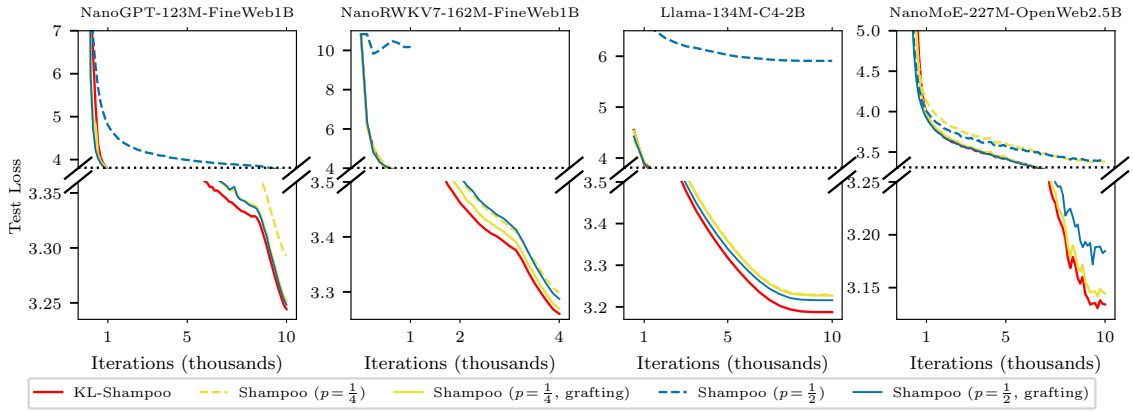

Figure 1: Preliminary results (random search using 120 runs for each method, Appx. A) on language models demonstrate that KL-Shampoo removes the need for step-size grafting with Adam. Shampoo without grafting does not perform well. In particular, Shampoo with power $p = 1/2$ fails to train the RWKV7 model in all 120 runs when grafting is disabled.

independently and the other is fixed as the identity, which is known as a one-sided preconditioner [4, 28]. For preconditioning, the constant scalar is $1/\sqrt{d_a d_b}$, which could help align with the scaling of Adam. Here, we approximate the required expectations using a single sample [21] such as $\mathbb{E}[\boldsymbol{G}\boldsymbol{G}^\top] \approx \boldsymbol{G}\boldsymbol{G}^\top$ and $\mathbb{E}[\boldsymbol{G}^\top\boldsymbol{G}] \approx \boldsymbol{G}^\top\boldsymbol{G}$. This KL interpretation highlights a key **limitation** of Shampoo's estimation rule: it is not the optimal solution to the KL problem when both factors are learned jointly. This limitation motivates our improved schemes, which we introduce in Sec. 3.2.

Claim 1 ***(Shampoo's Kronecker-based covariance estimation)*** *The optimal solution of KL minimization* $\min_{\boldsymbol{S}_a} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big)$ *with a one-sided preconditioner* $\boldsymbol{S} = (1/d_b \boldsymbol{S}_a) \otimes \boldsymbol{I}_b$ *is* $\boldsymbol{S}_a^* = \mathbb{E}[\boldsymbol{G}\boldsymbol{G}^\top]$*, where* d_k *is the dimension of matrix* $\boldsymbol{S}_k \in \mathbb{R}^{d_k \times d_k}$ *for* $k \in \{a, b\}$ *and* $\boldsymbol{G} = \mathrm{Mat}(\boldsymbol{g})$.

Likewise, we can obtain the estimation rule for $\boldsymbol{S}_b$ *by considering* $\boldsymbol{S} = \boldsymbol{I}_a \otimes (1/d_a \boldsymbol{S}_b)$.

3.2. Improving Shampoo's estimation: Idealized KL-Shampoo

Our KL perspective reveals a key limitation of Shampoo's Kronecker factor estimation: this scheme does not adequately solve the KL minimization problem when both factors are learned jointly. Motivated by this observation, we design an improved estimation rule that updates the two factors simultaneously. We refer to this scheme as *idealized KL-Shampoo*.

Claim 2 ***(Idealized KL-Shampoo's covariance estimation for*** $\boldsymbol{S}_a$ ***and*** $\boldsymbol{S}_b$***)*** *The optimal solution of KL minimization* $\min_{\boldsymbol{S}_a, \boldsymbol{S}_b} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big)$ *with a two-sided precontioner* $\boldsymbol{S} = \boldsymbol{S}_a \otimes \boldsymbol{S}_b$ *is*

$$\boldsymbol{S}_a^* = \frac{1}{d_b}\mathbb{E}[\boldsymbol{G}\big(\boldsymbol{S}_b^*\big)^{-1}\boldsymbol{G}^\top], \quad \boldsymbol{S}_b^* = \frac{1}{d_a}\mathbb{E}[\boldsymbol{G}^\top\big(\boldsymbol{S}_a^*\big)^{-1}\boldsymbol{G}]. \tag{3}$$

Idealized KL-Shampoo's estimation Claim 2 establishes a closed-form expression (see Eq. (3)) when simultaneously learning both Kronecker factors to minimize the KL problem. This expression was originally considered as a theoretical example [17, 18] for covariance estimation and later, Vyas et al. [26] consider a similar expression based on a heuristic motivated by gradient whitening.

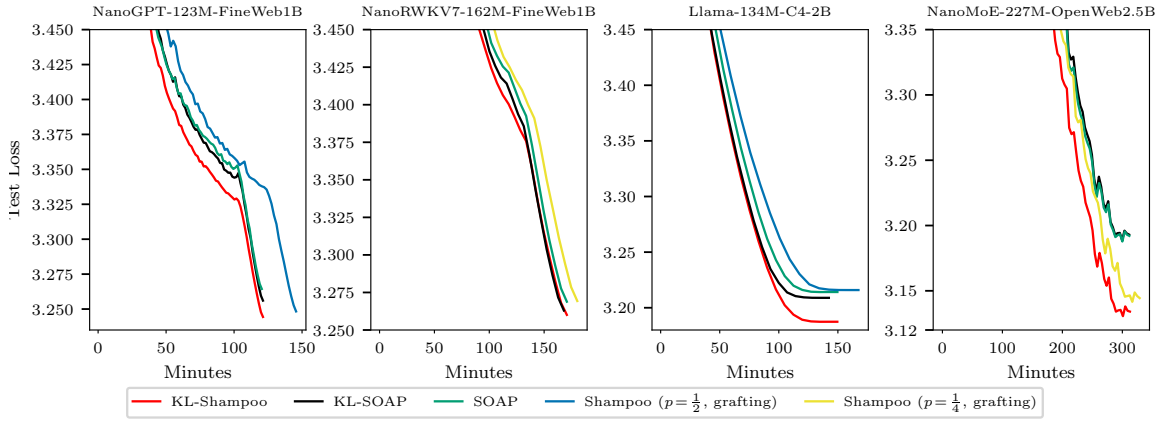

Figure 2: Preliminary results (random search using 120 runs for each method, Appx. A) on language models demonstrate KL-Shampoo meets or exceeds SOAP's efficiency using QR decomposition. We also include the best Shampoo run in the plots for completeness.

However, we cannot directly apply this expression due to the correlated update between $\boldsymbol{S}_a$ and $\boldsymbol{S}_b$. For example, solving $\boldsymbol{S}_a^*$ requires knowing $\boldsymbol{S}_b^*$ in Eq. (3) or vice versa. To overcome this, we use an estimated $\boldsymbol{S}_k$ to approximate $\boldsymbol{S}_k^*$ for $k \in \{a, b\}$ and propose a moving average scheme:

$$\boldsymbol{S}_a \leftarrow (1-\beta_2)\boldsymbol{S}_a + \frac{\beta_2}{d_b}\mathbb{E}[\boldsymbol{G}\boldsymbol{S}_b^{-1}\boldsymbol{G}^\top], \quad \boldsymbol{S}_b \leftarrow (1-\beta_2)\boldsymbol{S}_b + \frac{\beta_2}{d_a}\mathbb{E}[\boldsymbol{G}^\top\boldsymbol{S}_a^{-1}\boldsymbol{G}]. \tag{4}$$

We can justify this scheme and establish a formal connection to the theoretical approach of Lin et al. [17, 18] using the proximal-gradient framework [16]. Notably, our approach uses $\boldsymbol{S}^{-1/2}$ for preconditioning (Eq. (1)), following Shampoo, whereas Lin et al. [17, 18] propose using $\boldsymbol{S}^{-1}$. A straightforward implementation of our scheme is computationally expensive, since it requires additional matrix inversions (highlighted in red in Eq. (4)) as well as the slow eigendecomposition needed for Shampoo-type preconditioning (e.g., $\boldsymbol{S}^{-1/2}$). However, these issues can be alleviated—in the next section we propose a computationally efficient implementation of our method.

4. Efficient Implementation: KL-Shampoo with QR Decomposition

The eigendecomposition used in Shampoo's implementation [24] is more computationally expensive than QR decomposition [25]. Motivated by this observation, we aim to improve KL-Shampoo's computational efficiency by replacing the eigendecomposition with QR decomposition. However, incorporating QR decomposition into KL-Shampoo is non-trivial because the eigenvalues of the Kronecker factors are required, and QR does not provide them. Specifically, the eigenvalues are essential for a reduction in the computational cost of KL-Shampoo in two reasons: (1) they remove the need to compute the matrix $-1/2$ power, $\boldsymbol{S}^{-1/2} = (\boldsymbol{Q}_a \mathrm{Diag}(\boldsymbol{\lambda}_a^{\odot -1/2})\boldsymbol{Q}_a^\top) \otimes (\boldsymbol{Q}_b \mathrm{Diag}(\boldsymbol{\lambda}_b^{\odot -1/2})\boldsymbol{Q}_b^\top)$, used for KL-Shampoo's preconditioning; (2) they also eliminate expensive matrix inversions in its Kronecker estimation scheme (Eq. (4)), such as $\boldsymbol{S}_b^{-1}$ in the update for $\boldsymbol{S}_a$:

$$\boldsymbol{S}_a \leftarrow (1-\beta_2)\boldsymbol{S}_a + \frac{\beta_2}{d_b}\mathbb{E}[\boldsymbol{G}\boldsymbol{S}_b^{-1}\boldsymbol{G}^\top] = (1-\beta_2)\boldsymbol{S}_a + \frac{\beta_2}{d_b}\mathbb{E}[\boldsymbol{G}\boldsymbol{Q}_b \mathrm{Diag}(\boldsymbol{\lambda}_b^{\odot -1})\boldsymbol{Q}_b^\top \boldsymbol{G}^\top], \tag{5}$$

where $\boldsymbol{Q}_k$ and $\boldsymbol{\lambda}_k$ are eigenbasis and eigenvalues of $\boldsymbol{S}_k$ for $k \in \{a, b\}$, respectively.

Shampoo with power $p = 1/2$ versus Our idealized KL-Shampoo

1: Gradient Computation $\boldsymbol{g} := \nabla \ell(\boldsymbol{\theta})$
$\boldsymbol{G} := \mathrm{Mat}(\boldsymbol{g}) \in \mathbb{R}^{d_a \times d_b}$
2: Covariance Estimation (each iter)
$\begin{pmatrix} \boldsymbol{S}_a \\ \boldsymbol{S}_b \end{pmatrix} \leftarrow (1-\beta_2) \begin{pmatrix} \boldsymbol{S}_a \\ \boldsymbol{S}_b \end{pmatrix} + \beta_2 \begin{pmatrix} \Delta_a \\ \Delta_b \end{pmatrix}$
$\Delta_a := \begin{cases} \boldsymbol{G}\boldsymbol{G}^\top \\ \boldsymbol{G}\boldsymbol{Q}_b \mathrm{Diag}(\boldsymbol{\lambda}_b^{\odot -1}) \boldsymbol{Q}_b^\top \boldsymbol{G}^\top / d_b \end{cases}$
$\Delta_b := \begin{cases} \boldsymbol{G}^\top \boldsymbol{G} \\ \boldsymbol{G}^\top \boldsymbol{Q}_a \mathrm{Diag}(\boldsymbol{\lambda}_a^{\odot -1}) \boldsymbol{Q}_a^\top \boldsymbol{G} / d_a \end{cases}$
3: Eigen Decomposition (every $T \geq 1$ iters)
$\boldsymbol{\lambda}_k, \boldsymbol{Q}_k \leftarrow \mathrm{eig}(\boldsymbol{S}_k)$ for $k \in \{a, b\}$
4: Preconditioning using $\boldsymbol{Q} := \boldsymbol{Q}_a \otimes \boldsymbol{Q}_b$
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma(\boldsymbol{Q}\,\mathrm{Diag}(\boldsymbol{\lambda}_a \otimes \boldsymbol{\lambda}_b)^{-1/2} \boldsymbol{Q}^\top)\boldsymbol{g}$

Replacing the slow eigen decomposition with more efficient QR updates *(replaces Step 3)*

3a: **Frequent** Eigenvalue Estimation with Moving Average (each iter)
$\begin{pmatrix} \boldsymbol{\lambda}_a \\ \boldsymbol{\lambda}_b \end{pmatrix} \leftarrow (1-\beta_2) \begin{pmatrix} \boldsymbol{\lambda}_a \\ \boldsymbol{\lambda}_b \end{pmatrix} + \beta_2 \begin{pmatrix} \mathrm{diag}(\boldsymbol{Q}_a^\top \Delta_a \boldsymbol{Q}_a) \\ \mathrm{diag}(\boldsymbol{Q}_b^\top \Delta_b \boldsymbol{Q}_b) \end{pmatrix}$
3b: Infrequent Eigenbasis Estimation using QR (every $T \geq 1$ iters)
$\boldsymbol{Q}_k \leftarrow \mathrm{qr}(\boldsymbol{S}_k \boldsymbol{Q}_k)$ for $k \in \{a, b\}$

Figure 3: *Left:* Simplified update schemes without momentum, damping, and weight decay. *Right:* For computational efficiency, we replace the eigen step with our eigenvalue estimation and infrequent eigenbasis estimation using QR, where we estimate eigenvalues $\boldsymbol{\lambda}_k$ using an outdated eigenbasis $\boldsymbol{Q}_k$ for $k \in \{a, b\}$ and use the QR procedure.

KL-based estimation rule for the eigenvalues $\boldsymbol{\lambda}_a$ and $\boldsymbol{\lambda}_b$ using an outdated eigenbasis We aim to estimate the eigenvalues using an outdated eigenbasis to replace the slow eigendecomposition with a fast QR decomposition in KL-Shampoo. Eschenhagen et al. [11] propose estimating the eigenvalues from a Frobenius-norm perspective, using $\boldsymbol{\lambda}_k^{(\mathrm{Frob})} := \mathrm{diag}(\boldsymbol{Q}_k^\top \boldsymbol{S}_k \boldsymbol{Q}_k)$ for $k \in \{a, b\}$. However, our empirical results indicate that this approach becomes suboptimal when an outdated eigenbasis $\boldsymbol{Q}_k$ is reused to reduce the frequency and cost of QR decompositions. In contrast, our KL perspective (Claim 3) provides a principled alternative, allowing us to use an outdated eigenbasis. Building on this insight, we introduce a moving-average scheme (Fig. 3) for eigenvalue estimation, which can be justified through the proximal-gradient framework [16]. This allows us to update the eigenvalues *every iteration* while only updating the eigenbasis at a lower frequency via an efficient QR-based procedure, similar to SOAP. Since this scheme naturally scales the eigenvalues by the Kronecker factors' dimensions, according to Eschenhagen et al. [11], step-size grafting should not be necessary for KL-Shampoo, which we confirm empirically (Appx. A).

Claim 3 ***(Estimation rule for eigenvalues $\boldsymbol{\lambda}_a$ and $\boldsymbol{\lambda}_b$)*** *The optimal solution of KL minimization* $\min_{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_b} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big)$ *with preconditioner* $\boldsymbol{S} = (\boldsymbol{Q}_a \mathrm{Diag}(\boldsymbol{\lambda}_a) \boldsymbol{Q}_a^\top) \otimes (\boldsymbol{Q}_b \mathrm{Diag}(\boldsymbol{\lambda}_b) \boldsymbol{Q}_b^\top)$ *is*

$$\boldsymbol{\lambda}_a^* = \frac{1}{d_b} \mathrm{diag}\big(\boldsymbol{Q}_a^\top \mathbb{E}[\boldsymbol{G}\boldsymbol{P}_b^* \boldsymbol{G}^\top] \boldsymbol{Q}_a\big), \quad \boldsymbol{\lambda}_b^* = \frac{1}{d_a} \mathrm{diag}\big(\boldsymbol{Q}_b^\top \mathbb{E}[\boldsymbol{G}^\top \boldsymbol{P}_a^* \boldsymbol{G}] \boldsymbol{Q}_b\big), \tag{6}$$

where $\boldsymbol{P}_k^* := \boldsymbol{Q}_k \mathrm{Diag}\left((\boldsymbol{\lambda}_k^*)^{\odot -1}\right) \boldsymbol{Q}_k^\top$ *is considered as an approximation of* $\boldsymbol{S}_k^{-1}$ *for* $k \in \{a, b\}$ *when using an outdated eigenbasis* $\boldsymbol{Q} = \boldsymbol{Q}_a \otimes \boldsymbol{Q}_b$ *precomputed by QR.*

5. Conclusion

We introduced a KL perspective for interpreting Shampoo's structured second-moment estimation scheme. This perspective uncovers a previously unrecognized limitation, motivates an alternative estimation strategy to overcome it, enables a practical implementation of our approach, and extends naturally to tensor-valued estimation. Preliminary experiments demonstrate the effectiveness of our method and underscore its potential to further unlock Shampoo's performance.

Acknowledgements

The authors would like to thank Mubarak Shah (University of Central Florida), Frank Nielsen (Sony), Mark Schmidt (University of British Columbia), and Emtiyaz Khan (RIKEN) for their helpful feedback on the manuscript, and Cheng Wu (Microsoft) and Xiya Liu (Microsoft) for their insights on language models and their support. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, the companies sponsoring the Vector Institute (see www.vectorinstitute.ai/#partners), and Microsoft Azure ML compute.

References

[1] Naman Agarwal, Brian Bullins, Xinyi Chen, Elad Hazan, Karan Singh, Cyril Zhang, and Yi Zhang. Efficient full-matrix adaptive regularization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 102–110. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/agarwal19b.html.

[2] Naman Agarwal, Rohan Anil, Elad Hazan, Tomer Koren, and Cyril Zhang. Disentangling adaptive gradient methods from learning rates. *arXiv preprint arXiv:2002.11803*, 2020. doi:10.48550/arxiv.2002.11803.

[3] Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016. ISBN 9784431559788. doi:10.1007/978-4-431-55978-8.

[4] Kang An, Yuxing Liu, Rui Pan, Yi Ren, Shiqian Ma, Donald Goldfarb, and Tong Zhang. ASGO: Adaptive structured gradient optimization. *arXiv preprint arXiv:2503.20762*, 2025. doi:10.48550/arxiv.2503.20762.

[5] Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*, 2020. doi:10.48550/arxiv.2002.09018.

[6] Rajendra Bhatia. *Positive definite matrices*. Princeton University Press, 2007. ISBN 9780691129181. URL http://www.jstor.org/stable/j.ctt7rxv2.

[7] Peng Bo. RWKV-7: Surpassing GPT. https://github.com/BlinkDL/modded-nanogpt-rwkv, 2024. Accessed: 2025-06.

[8] George E Dahl, Frank Schneider, Zachary Nado, Naman Agarwal, Chandramouli Shama Sastry, Philipp Hennig, Sourabh Medapati, Runa Eschenhagen, Priya Kasimbeg, Daniel Suo, et al. Benchmarking neural network training algorithms. *arXiv preprint arXiv:2306.07179*, 2023. doi:10.48550/arxiv.2306.07179.

[9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

[10] Sai Surya Duvvuri, Fnu Devvrit, Rohan Anil, Cho-Jui Hsieh, and Inderjit S Dhillon. Combining axes preconditioners through Kronecker approximation for deep learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8j9hz8DVi8.

[11] Runa Eschenhagen, Aaron Defazio, Tsung-Hsien Lee, Richard E Turner, and Hao-Jun Michael Shi. Purifying Shampoo: Investigating Shampoo's heuristics by decomposing its preconditioner. *arXiv preprint arXiv:2506.03595*, 2025. doi:10.48550/arxiv.2506.03595.

[12] Athanasios Glentis. A minimalist optimizer design for LLM pretraining. https://github.com/OptimAI-Lab/Minimalist_LLM_Pretraining, 2025. Accessed: 2025-06.

[13] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

[14] Keller Jordan. NanoGPT (124M) in 3 minutes. https://github.com/KellerJordan/modded-nanogpt, 2024. Accessed: 2025-06.

[15] Priya Kasimbeg, Frank Schneider, Runa Eschenhagen, Juhan Bae, Chandramouli Shama Sastry, Mark Saroufim, Boyuan Fend, Less Wright, Edward Z Yang, Zachary Nado, et al. Accelerating neural network training: An analysis of the AlgoPerf competition. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CtM5xjRSfm.

[16] Mohammad Emtiyaz Khan, Reza Babanezhad, Wu Lin, Mark Schmidt, and Masashi Sugiyama. Faster stochastic variational inference using proximal-gradient methods with general divergence functions. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 319–328. AUAI Press, 2016. URL https://www.auai.org/uai2016/proceedings/papers/218.pdf.

[17] Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3992–4002. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lin19b.html.

[18] Wu Lin, Felix Dangel, Runa Eschenhagen, Juhan Bae, Richard E Turner, and Alireza Makhzani. Can we remove the square-root in adaptive gradient methods? A second-order perspective. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 29949–29973. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lin24e.html.

[19] Karl Löwner. Über monotone matrixfunktionen. *Mathematische Zeitschrift*, 38(1):177–216, 1934. doi:10.1007/BF01170633.

[20] Hà Quang Minh and Vittorio Murino. Covariances in computer vision and machine learning. *Synthesis Lectures on Computer Vision*, 7(4):1–170, 2017. ISSN 2153-1056. doi:10.1007/978-3-031-01820-6.

[21] Depen Morwani, Itai Shapira, Nikhil Vyas, Eran Malach, Sham M Kakade, and Lucas Janson. A new perspective on Shampoo's preconditioner. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=c6zI3Cp8c6.

[22] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer Cham, 2018. ISBN 9783319915784. doi:10.1007/978-3-319-91578-4.

[23] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, Jan 2006. ISSN 1573-1405. doi:10.1007/s11263-005-3222-z.

[24] Hao-Jun Michael Shi, Tsung-Hsien Lee, Shintaro Iwasaki, Jose Gallego-Posada, Zhijing Li, Kaushik Rangadurai, Dheevatsa Mudigere, and Michael Rabbat. A distributed data-parallel PyTorch implementation of the distributed Shampoo optimizer for training neural networks at-scale. *arXiv preprint arXiv:2309.06497*, 2023. doi:10.48550/arxiv.2309.06497.

[25] Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IDxZhXrpNf.

[26] Nikhil Vyas, Rosie Zhao, Depen Morwani, Mujin Kwun, and Sham Kakade. Improving SOAP using iterative whitening and Muon. https://nikhilvyas.github.io/SOAP_Muon.pdf, 2025.

[27] Cameron R. Wolfe. An extension of the NanoGPT repository for training small MOE models. https://github.com/wolfecameron/nanoMoE, 2025. Accessed: 2025-06.

[28] Shuo Xie, Tianhao Wang, Sashank J Reddi, Sanjiv Kumar, and Zhiyuan Li. Structured preconditioners in adaptive optimization: A unified analysis. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=GzS6b5Xvvu.

Appendix A. Experimental Setup and Preliminary Results

We evaluate KL-Shampoo on four language models based on existing implementations: NanoGPT [14] (123 M), NanoRWKV7 [7] (162 M), Llama [12] (134 M), and NanoMoE [27] (227 M). We consider NanoMoE, as it contains many 3D weight tensors. This model provides a natural testbed for evaluating a tensor extension of KL-Shampoo, derived directly from our KL perspective. In doing so, we demonstrate that our approach retains the same flexibility as Shampoo in handling tensor-valued weights. We consider several strong baselines, including Shampoo with $p = 1/2$ and $p = 1/4$ powers using a state-of-the-art implementation [24], and an improved variant of Shampoo: SOAP [25]. We train NanoGPT and NanoRWKV7 using a subset of FineWeb (1 B tokens), Llama using a subset of C4 (2 B tokens), and NanoMoE using a subset of OpenWebText (2.5 B tokens). All models except NanoMoE are trained using mini-batches with a batch size of 0.5 M. We use a batch size of 0.25 M to train NanoMoE to reduce the run time. We use the default step-size schedulers from the source implementations; NanoGPT and NanoRWKV7: linear warmup + constant step-size + linear cooldown; Llama and NanoMoE: linear warmup + cosine step-size. We tune all available hyperparameters of each method using random search with 120 runs. In our experiments, Shampoo performs eigendecomposition every 10 steps, while KL-Shampoo and SOAP perform QR decomposition every 10 steps. Preliminary results demonstrate KL-Shampoo outperforms Shampoo (Fig. 1) without needing grafting, and outperforms SOAP while matching its efficiency (Fig. 2).

Appendix B. Proof of Claim 1

We will show that the optimal solution of KL minimization $\min_{\boldsymbol{S}_a} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big)$ with a one-sided preconditioner $\boldsymbol{S} = (1/d_b \boldsymbol{S}_a) \otimes \boldsymbol{I}_b$ is $\boldsymbol{S}_a^* = \mathbb{E}[\boldsymbol{G}\boldsymbol{G}^\top]$.

By definition in Eq. (2) and substituting $\boldsymbol{S} = (1/d_b \boldsymbol{S}_a) \otimes \boldsymbol{I}_b$, we can simplify the objective function as

$$
\begin{aligned}
&\mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big) \\
&= \frac{1}{2}\big(\log\det(\boldsymbol{S}) + \mathrm{Tr}(\boldsymbol{S}^{-1}\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top])\big) + \text{const.} \\
&= \frac{1}{2}\big(d_b \log\det(\frac{1}{d_b}\boldsymbol{S}_a) + \mathrm{Tr}(\boldsymbol{S}^{-1}\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top])\big) + \text{const.} \text{ (Kronecker identity for matrix determinant)} \\
&= \frac{1}{2}\big(d_b \log\det(\boldsymbol{S}_a) + \mathrm{Tr}(\boldsymbol{S}^{-1}\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top])\big) + \text{const.} \text{ (identity for a log-determinant)} \\
&= \frac{1}{2}\big(d_b \log\det(\boldsymbol{S}_a) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{S}^{-1}\boldsymbol{g}\boldsymbol{g}^\top)]\big) + \text{const.} \text{ (linearity of the expectation)} \\
&= \frac{1}{2}\big(d_b \log\det(\boldsymbol{S}_a) + \mathbb{E}[\mathrm{Tr}(d_b \boldsymbol{S}_a^{-1}\boldsymbol{G}\boldsymbol{I}_b\boldsymbol{G}^\top)]\big) + \text{const.} \text{ (identity for a Kronecker vector product)} \\
&= \frac{d_b}{2}\big(\log\det(\boldsymbol{S}_a) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{S}_a^{-1}\boldsymbol{G}\boldsymbol{G}^\top)]\big) + \text{const.} \\
&= \frac{d_b}{2}\big(-\log\det(\boldsymbol{P}_a) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{P}_a\boldsymbol{G}\boldsymbol{G}^\top)]\big) + \text{const.},
\end{aligned}
$$

where $\boldsymbol{G} = \mathrm{Mat}(\boldsymbol{g})$ and $\boldsymbol{P}_a := \boldsymbol{S}_a^{-1}$.

If we achieve the optimal solution, the gradient stationary condition must be satisfied regardless of the gradient with respect to $\boldsymbol{S}_a$ or $\boldsymbol{S}_a^{-1} \equiv \boldsymbol{P}_a$, such as

$$\begin{aligned}
\boldsymbol{0} &= \partial_{\boldsymbol{S}_a^{-1}} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big) \\
&= \partial_{\boldsymbol{P}_a} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big) \\
&= \frac{d_b}{2}\big(- \boldsymbol{P}_a^{-1} + \mathbb{E}[\boldsymbol{G}\boldsymbol{G}^\top]\big) \quad \text{(matrix calculus identities)} \\
&= \frac{d_b}{2}\big(- \boldsymbol{S}_a + \mathbb{E}[\boldsymbol{G}\boldsymbol{G}^\top]\big).
\end{aligned}$$

Thus, the optimal solution must be $\boldsymbol{S}_a^* = \mathbb{E}[\boldsymbol{G}\boldsymbol{G}^\top]$ to satisfy this stationary condition.

Appendix C. Proof of Claim 2

We will show that the optimal solution of KL minimization $\min_{\boldsymbol{S}_a, \boldsymbol{S}_b} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big)$ with a two-sided preconditioner $\boldsymbol{S} = \boldsymbol{S}_a \otimes \boldsymbol{S}_b$ is $\boldsymbol{S}_a^* = \frac{1}{d_b}\mathbb{E}[\boldsymbol{G}\big(\boldsymbol{S}_b^*\big)^{-1}\boldsymbol{G}^\top]$ and $\boldsymbol{S}_b^* = \frac{1}{d_a}\mathbb{E}[\boldsymbol{G}^\top\big(\boldsymbol{S}_a^*\big)^{-1}\boldsymbol{G}]$.

Similar to the proof of Claim 1 in Appx. B, we can simplify the objective function as

$$\begin{aligned}
&\mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big) \\
&= \frac{1}{2}\big(\log\det(\boldsymbol{S}) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{S}^{-1}\boldsymbol{g}\boldsymbol{g}^\top)]\big) + \text{const.} \\
&= \frac{1}{2}\big(d_b \log\det(\boldsymbol{S}_a) + d_a \log\det(\boldsymbol{S}_b) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{S}^{-1}\boldsymbol{g}\boldsymbol{g}^\top)]\big) + \text{const.} \text{ (identity for a log-determinant)} \\
&= \frac{1}{2}\big(d_b \log\det(\boldsymbol{S}_a) + d_a \log\det(\boldsymbol{S}_b) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{S}_a^{-1}\boldsymbol{G}\boldsymbol{S}_b^{-1}\boldsymbol{G}^\top)]\big) + \text{const.} \text{ (identity for a Kronecker-vector-product)} \\
&= \frac{1}{2}\big(- d_b \log\det(\boldsymbol{P}_a) - d_a \log\det(\boldsymbol{P}_b) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{P}_a\boldsymbol{G}\boldsymbol{P}_b\boldsymbol{G}^\top)]\big) + \text{const.},
\end{aligned}$$

where $\boldsymbol{P}_k := \boldsymbol{S}_k^{-1}$ for $k \in \{a, b\}$.

The optimal solution must satisfy the gradient stationarity condition with respect to $\{\boldsymbol{S}_a, \boldsymbol{S}_b\}$. Notice that the gradient with respect to $\{\boldsymbol{S}_a^{-1}, \boldsymbol{S}_b^{-1}\}$ can be expressed in terms of the gradient with respect to $\{\boldsymbol{S}_a, \boldsymbol{S}_b\}$ as $\partial_{\boldsymbol{S}_a^{-1}}\mathrm{KL} = -\boldsymbol{S}_a\big(\partial_{\boldsymbol{S}_a}\mathrm{KL}\big)\boldsymbol{S}_a$ and $\partial_{\boldsymbol{S}_b^{-1}}\mathrm{KL} = -\boldsymbol{S}_b\big(\partial_{\boldsymbol{S}_b}\mathrm{KL}\big)\boldsymbol{S}_b$. Thus, the optimal solution must satisfy the following gradient stationarity condition with respect to $\{\boldsymbol{S}_a^{-1}, \boldsymbol{S}_b^{-1}\}$.

$$0 = \partial_{\boldsymbol{S}_a^{-1}} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big); \quad 0 = \partial_{\boldsymbol{S}_b^{-1}} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big).$$

Simplifying the left expression

$$\begin{aligned}
0 &= \partial_{\boldsymbol{S}_a^{-1}} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big) \\
&= \partial_{\boldsymbol{P}_a} \mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top], \boldsymbol{S}\big) \\
&= \frac{1}{2}\big(- d_b \boldsymbol{P}_a^{-1} + \mathbb{E}[\boldsymbol{G}\boldsymbol{P}_b\boldsymbol{G}^\top]\big)
\end{aligned}$$

gives us this equation

$$0 = \frac{1}{2}(-d_b \boldsymbol{S}_a^* + \mathbb{E}[\boldsymbol{G}\big(\boldsymbol{S}_b^*\big)^{-1}\boldsymbol{G}^\top])$$

that the optimal solution must satisfy.

This naturally leads to the following expression:

$$\boldsymbol{S}_a^* = \frac{1}{d_b}\mathbb{E}[\boldsymbol{G}\big(\boldsymbol{S}_b^*\big)^{-1}\boldsymbol{G}^\top].$$

Likewise, we can obtain the following expression by simplifying the right expression of the gradient stationary condition.

$$\boldsymbol{S}_b^* = \frac{1}{d_a}\mathbb{E}[\boldsymbol{G}^\top\big(\boldsymbol{S}_a^*\big)^{-1}\boldsymbol{G}].$$

Appendix D. Proof of Claim 3

We will show that the optimal solution of KL minimization $\min_{\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_b}\mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top],\boldsymbol{S}\big)$ with a two-sided preconditioner $\boldsymbol{S} = (\boldsymbol{Q}_a\mathrm{Diag}(\boldsymbol{\lambda}_a)\boldsymbol{Q}_a^\top)\otimes(\boldsymbol{Q}_b\mathrm{Diag}(\boldsymbol{\lambda}_b)\boldsymbol{Q}_b^\top)$ is $\boldsymbol{\lambda}_a^* = \frac{1}{d_b}\mathrm{diag}\big(\boldsymbol{Q}_a^\top\mathbb{E}[\boldsymbol{G}\boldsymbol{P}_b^*\boldsymbol{G}^\top]\boldsymbol{Q}_a\big)$ and $\boldsymbol{\lambda}_b^* = \frac{1}{d_a}\mathrm{diag}\big(\boldsymbol{Q}_b^\top\mathbb{E}[\boldsymbol{G}^\top\boldsymbol{P}_a^*\boldsymbol{G}]\boldsymbol{Q}_b\big)$, where $\boldsymbol{P}_k^* := \boldsymbol{Q}_k\mathrm{Diag}\left((\boldsymbol{\lambda}_k^*)^{\odot-1}\right)\boldsymbol{Q}_k^\top$, and $\boldsymbol{Q}_k$ is known and precomputed by $\mathtt{QR}$ for $k\in\{a,b\}$.

Let $\boldsymbol{S}_k := \boldsymbol{Q}_k\mathrm{Diag}(\boldsymbol{\lambda}_k)\boldsymbol{Q}_k^\top$ for $k\in\{a,b\}$. Because $\boldsymbol{Q}_k$ is orthogonal, it is easy to see that $\boldsymbol{S}_k^{-1} := \boldsymbol{Q}_k\mathrm{Diag}(\big(\boldsymbol{\lambda}_k\big)^{\odot-1})\boldsymbol{Q}_k^\top$.

Similar to the proof of Claim 2 in Appx. C, we can simplify the following objective function by substituting $\boldsymbol{S}_a$ and $\boldsymbol{S}_b$. Here, we also utilize the the orthogonality of $\boldsymbol{Q}_k$.

$$\begin{aligned}
&\mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top],\boldsymbol{S}\big)\\
&\quad= \frac{1}{2}\big(d_b\log\det(\boldsymbol{S}_a) + d_a\log\det(\boldsymbol{S}_b) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{S}_a^{-1}\boldsymbol{G}\boldsymbol{S}_b^{-1}\boldsymbol{G}^\top)]\big) + \text{const.}\\
&\quad= \frac{1}{2}\big(d_b\log\det(\boldsymbol{Q}_a\mathrm{Diag}(\boldsymbol{\lambda}_a)\boldsymbol{Q}_a^\top) + d_a\log\det(\boldsymbol{Q}_b\mathrm{Diag}(\boldsymbol{\lambda}_b)\boldsymbol{Q}_b^\top) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{S}_a^{-1}\boldsymbol{G}\boldsymbol{S}_b^{-1}\boldsymbol{G}^\top)]\big) + \text{const.}\\
&\quad= \frac{1}{2}\big((d_b\sum_i\log(\lambda_a^{(i)})) + (d_a\sum_j\log(\lambda_b^{(j)})) + \mathbb{E}[\mathrm{Tr}(\boldsymbol{Q}_a\mathrm{Diag}(\boldsymbol{\lambda}_a^{\odot-1})\boldsymbol{Q}_a^\top\boldsymbol{G}\boldsymbol{Q}_b\mathrm{Diag}(\boldsymbol{\lambda}_b^{\odot-1})\boldsymbol{Q}_b^\top\boldsymbol{G}^\top)]\big) + \text{const.}
\end{aligned}$$

The optimal $\boldsymbol{\lambda}_a$ and $\boldsymbol{\lambda}_b$ should satisfy the gradient stationary condition.

$$\begin{aligned}
0 &= \partial_{\boldsymbol{\lambda}_a}\mathrm{KL}\big(\mathbb{E}[\boldsymbol{g}\boldsymbol{g}^\top],\boldsymbol{S}\big)\\
&= \frac{1}{2}\big(d_b\boldsymbol{\lambda}_a^{\odot-1} + \partial_{\boldsymbol{\lambda}_a}\mathbb{E}[\mathrm{Tr}(\boldsymbol{Q}_a\mathrm{Diag}(\boldsymbol{\lambda}_a^{\odot-1})\boldsymbol{Q}_a^\top\boldsymbol{G}\overbrace{\boldsymbol{Q}_b\mathrm{Diag}(\boldsymbol{\lambda}_b^{\odot-1})\boldsymbol{Q}_b^\top}^{=\boldsymbol{P}_b}\boldsymbol{G}^\top)]\big)\\
&= \frac{1}{2}\big(d_b\boldsymbol{\lambda}_a^{\odot-1} + \partial_{\boldsymbol{\lambda}_a}\mathbb{E}[\mathrm{Tr}(\mathrm{Diag}(\boldsymbol{\lambda}_a^{\odot-1})\boldsymbol{Q}_a^\top\boldsymbol{G}\boldsymbol{P}_b\boldsymbol{G}^\top\boldsymbol{Q}_a)]\big)\\
&= \frac{1}{2}\big(d_b\boldsymbol{\lambda}_a^{\odot-1} + \partial_{\boldsymbol{\lambda}_a}\mathbb{E}[\boldsymbol{\lambda}_a^{\odot-1}\odot\mathrm{diag}\big(\boldsymbol{Q}_a^\top\boldsymbol{G}\boldsymbol{P}_b\boldsymbol{G}^\top\boldsymbol{Q}_a\big)]\big) \quad \text{(utilize the trace and the diagonal structure)}\\
&= \frac{1}{2}\big(d_b\boldsymbol{\lambda}_a^{\odot-1} - \mathbb{E}[\boldsymbol{\lambda}_a^{\odot-2}\odot\mathrm{diag}\big(\boldsymbol{Q}_a^\top\boldsymbol{G}\boldsymbol{P}_b\boldsymbol{G}^\top\boldsymbol{Q}_a\big)]\big)\\
&= \frac{1}{2}\big(d_b\boldsymbol{\lambda}_a^{\odot-1} - \boldsymbol{\lambda}_a^{\odot-2}\odot\mathrm{diag}\big(\boldsymbol{Q}_a^\top\mathbb{E}[\boldsymbol{G}\boldsymbol{P}_b\boldsymbol{G}^\top]\boldsymbol{Q}_a\big)\big)\\
\iff 0 &= d_b\boldsymbol{\lambda}_a - \mathrm{diag}\big(\boldsymbol{Q}_a^\top\mathbb{E}[\boldsymbol{G}\boldsymbol{P}_b\boldsymbol{G}^\top]\boldsymbol{Q}_a\big)\big)
\end{aligned}$$

We obtain the optimal solution by solving this equation.

$$\boldsymbol{\lambda}_a^* = \frac{1}{d_b}\mathrm{diag}\big(\boldsymbol{Q}_a^\top \mathbb{E}[\boldsymbol{G}\boldsymbol{P}_b^*\boldsymbol{G}^\top]\boldsymbol{Q}_a)\big)$$

Similarly, we can obtain the other expression.